



Uma Estratégia para Melhoria de Processo de Desenvolvimento de Software Baseado em Componentes

Alfredo N. Tsukumo¹, Rafael V. M. dos Santos², Wesley Marinho², Leandro de P. Silva², Clênio F. Salviano¹

¹DMPS-CenPRA – Divisão de Melhoria de Processo de Software do
Centro de Pesquisas Renato Archer
Rod. D. Pedro I km 143,6 – 13069-901 Campinas – Brasil

²SWQuality Consultoria e Sistemas LTDA
Campus Histórico da Ufla - 37200-000 - Lavras – Brasil
{alfredo.tsukumo;clenio.salviano}@cenpra.gov.br
{rafael;leandro;weslei}@swquality.com.br

Abstract: *This paper describes a strategy for the specialization of software process capability models for the specific context of component based software development process improvement. As an example of the application of this strategy, an extension of CMMI-SE/SW v1.1 model is described with the specialization of some of its process areas and the addition of three new process areas corresponding to the processes from Reuse Process Group of the ISO/IEC 15504-5 model. The strategy is based in the PRO2PI approach for using process capability profiles to process improvement. The example application was done in the projects COMP-GOV and FLO-PREF.*

Resumo: *Este artigo descreve uma estratégia para a especialização de modelos de capacidade de processo de software para o contexto específico da melhoria de processo de desenvolvimento de software baseado em componentes. Como exemplo, é descrita a utilização desta estratégia para extensão do modelo CMMI-SE/SW v1.1 com a especialização de algumas de suas áreas de processo e a inclusão dos processos do grupo de reuso da ISO/IEC 15504-5. A estratégia para especialização é baseada na abordagem PRO2PI para o uso de perfis de capacidade de processo para melhoria de processo. A utilização exemplo é realizada como parte dos projetos COMP-GOV e FLO-PREF.*

1. Introdução

A Melhoria de Processo de Software (MPS) baseada em Modelos de Capacidade de Processo em geral e, mais particularmente nos níveis de maturidade do modelo SW-CMM foi estabelecida nos anos 1990 como uma abordagem prática e eficiente para melhorar o software. Além do SW-CMM, outros modelos foram desenvolvidos e utilizados para a MPS, como, por exemplo, o modelo CMMI-SE/SW [Chrissis et al. 2004], que aprimorou, ampliou e substituiu o SW-CMM, o modelo da ISO/IEC 15504-5 [2006], o modelo iCMM [Ibrahim et al. 2001] e o modelo MR-MPS [SOFTEX 2005].



Nesse quadro de multiplicação de modelos de capacidade de processo, a ISO/IEC 15504-2 foi desenvolvida como um *framework* de modelos para avaliação de processos para a melhoria de processos e determinação da capacidade [ISO/IEC 15504-2 2003]. Nesse sentido, modelos diferentes podem ser desenvolvidos (especializados) para diferentes segmentos e domínios, sob a mesma estrutura por ela definida. Muitos modelos relevantes atuais de capacidade de processo seguem, em maior ou menor grau, esta estrutura, incluindo os citados acima.

Há uma forte tendência à especialização de modelos para aplicação em ambientes e contextos específicos diversos, cobrindo desde domínios de aplicação (ex. setor bancário, setor automobilístico, automação industrial, etc.) até modelos de desenvolvimento de software como neste caso, o baseado em componentes (CBD¹). O *framework* da ISO/IEC 15504-2 fornece as bases para a sua realização.

O Desenvolvimento de Software Baseado em Componentes (DSBC) é a resposta atual à velha questão do reuso de software. Como diz documento do Ministério de Ciência e Tecnologia: “A principal motivação para o uso de componentes é a potencialização do reuso de software e o conseqüente ganho de produtividade que isto proporciona. Esta questão, entretanto, já freqüenta os conceitos de engenharia de software há décadas. A ESBC [Engenharia de Software Baseada em Componentes] é um passo a frente no reuso de código, ao pregar o reuso “*as is*”, ou seja, o componente é reutilizado sem alterar sua implementação, sem custos de desenvolvimento, apenas de montagem”[MCT 2005].

Este artigo se insere dentro desse panorama e descreve:

- uma estratégia para a especialização de modelos de capacidade de processo de software para o contexto específico da melhoria de processo de desenvolvimento de software baseado em componentes, e
- uma utilização exemplo desta estratégia para extensão do modelo CMMI-SE/SW v1.1 com a especialização de algumas de suas áreas de processo e a inclusão dos processos do grupo de reuso da ISO/IEC 15504-5.

A estratégia é baseada na abordagem PRO2PI. A utilização exemplo é realizada para os projetos COMP-GOV e FLO-PREF.

O restante do artigo é organizado da seguinte forma. A Seção 2 apresenta a estratégia. A Seção 3 apresenta o contexto da aplicação, que são os requisitos de melhoria de processo de software dos projetos COMP-GOV e FLO-PREF. A Seção 4 apresenta uma utilização da estratégia. A Seção 5 apresenta a conclusão, que inclui um resumo do trabalho, considerações sobre a validação e a seqüência do trabalho.

2. Estratégia para Melhoria de Processo de Desenvolvimento de Software Baseado em Componentes

Esta estratégia é baseada na abordagem PRO2PI [Salviano 2006]. PRO2PI provê um apoio metodológico para a melhoria de processo orientada a um perfil de capacidade de processo que pode utilizar elementos de qualquer modelo de capacidade de processo.

¹ Utiliza-se o acrônimo CBD para Component Based Development, bastante difundido.



Este apoio metodológico também pode ser utilizado para definir modelos mais específicos para contextos de negócio de um segmento ou domínio. PRO2PI é composto por um conjunto de propriedades, um meta modelo, um ciclo de melhoria e um conjunto de medições.

A estratégia para melhoria de processo de desenvolvimento de software baseada em componentes é orientada pela definição de especializações dos modelos mais utilizados para engenharia de software em geral. Esta estratégia foca mais no uso destes modelos do que na definição de um novo modelo. Esta estratégia pode ser considerada também como uma estratégia baseada em componentes, à medida que ela preconiza a definição de novos componentes (no caso novas áreas de processo) e a especialização de componentes mais genéricos (no caso as áreas de processo dos modelos para engenharia de software).

Esta estratégia é orientada por sete decisões principais:

- a) revisar o estado da arte e estado da prática da área, no caso, Desenvolvimento de Software Baseado em Componentes;
- b) identificar o modelo de capacidade para a melhoria de processo da engenharia de software em geral, que seja mais apropriado para as características da especialização e utiliza-lo como a base para a especialização;
- c) identificar ou definir um conjunto de novas áreas de processo para cobrir os principais aspectos específicos da Engenharia de Software Baseada em Componentes;
- d) representar estas novas áreas de processo no mesmo formato do modelo identificado como base para a especialização;
- e) identificar as áreas de processo do modelo identificado com o base mais afetadas pela Engenharia de Software Baseada em Componentes e prover alterações com relações ao uso das mesmas.
- f) identificar processos genéricos mais relevantes e utiliza-lo como referência adicional para a especialização do modelo;
- g) considerar as práticas de organizações relevantes que já utilizam Engenharia de Software Baseada em Componentes e também utiliza-las como referência adicional para a especialização do modelo.
- h) utilizar a especialização em organizações de software, analisar os resultados e melhorar a especialização com base nestas análises.

Para CBD, o modelo mais significativo é o OOSPICE (*Software Process Improvement and Capability dEtermination for Object Oriented / Component Based Software Development*) [Stallinger et al. 2002]. Entre os objetivos principais do OOSPICE estão “um metamodelo unificado do processo CBD, uma metodologia de avaliação de CBD, perfis de capacidade de provedores de componentes, uma metodologia de CBD e extensões para padrão de avaliação de processo da ISO/IEC 15504”. OOSPICE é um modelo completo para determinação de capacidade específica para CDB.

A estratégia adotada pelo OOSPICE, de desenvolver um novo modelo, tem a deficiência de dificultar a adoção deste modelo por organizações que já utilizam outros



modelos. A estratégia proposta neste trabalho é orientada por reuso e pela composição com os modelos já utilizados.

3. Contexto da utilização exemplo da estratégia: Projetos COMP-GOV e FLO-PREF

O desenvolvimento do Modelo de Referência de Processo para Desenvolvimento baseado em componentes (PRM.CBD) é parte de dois projetos (COMP-GOV e FLO-PREF), para o desenvolvimento de uma biblioteca pública de componentes de software para aplicação no domínio de governo eletrônico. Ambos os projetos são financiados pela FINEP – Financiadora de Estudos e Projetos [FINEP 2004].

Os projetos COMP-GOV e FLO-PREF são realizados por dois consórcios que englobam empresas privadas, universidades e centros de pesquisa. Nos dois projetos, os consórcios são responsáveis por definir e desenvolver partes específicas do projeto como, por exemplo, um modelo de negócios sustentável, um modelo de componente, um repositório de componentes com o seu processo de gerência, processos de desenvolvimento **de** e **com** componentes, um modelo de capacidade de processo junto com um processo de avaliação para desenvolvimento baseado em componentes, um modelo de qualidade de componente e seu processo de avaliação e um conjunto de componentes e sistemas que os utilizam.

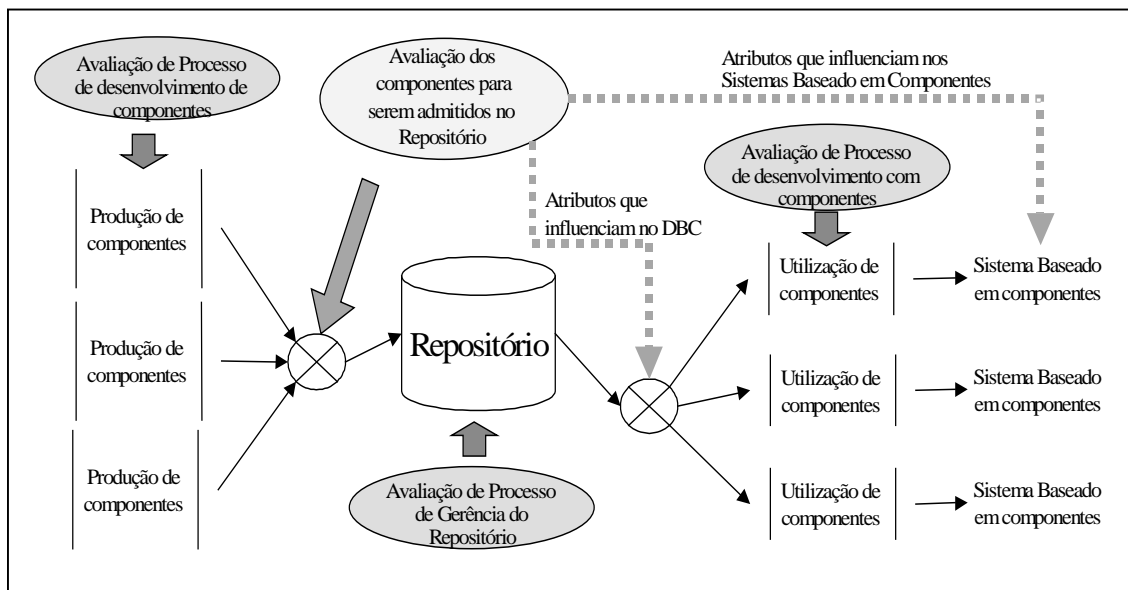


Figura 1 - Avaliações de Produto e Processo em desenvolvimento baseado em componentes

Nos dois projetos, o CenPRA é responsável pelo modelo de capacidade de processo para desenvolvimento baseado em componentes e pelo modelo de qualidade dos componentes e seu processo de avaliação, como mostra a Figura 1. Este artigo descreve apenas o desenvolvimento do Modelo de Capacidade de Processo a ser usado como referência para a melhoria e avaliações do processo de gerência do repositório e dos processos de desenvolvimento **de** e **com** componentes.



4. Utilização exemplo da estratégia

No contexto dos projetos COMP-GOV e FLO-PREF, essa estratégia foi aplicada com as seguintes diretrizes chaves:

- a) considerar o modelo CMMI-SE/SW versão 1,1, na sua representação contínua, como um modelo de referência de capacidade de processo para engenharia de software em geral, por causa de sua qualidade e sua disseminação na indústria de software. Os outros modelos estudados não foram escolhidos porque o ISO/IEC 15504-5 e o iCMM ainda não estão muito difundidos e o MR-MPS é um modelo estagiado.
- b) considerar os três processos do grupo de processos de reuso do modelo ISO/IEC 15504-5 como os componentes a serem incluídos para cobrir a principal área de Engenharia de Software Baseada em Componentes. O processo Engenharia de Domínio cobre os principais pontos sobre o processo de desenvolvimento de componentes. O processo Gerência do Programa de Reuso cobre os principais pontos sobre o processo de desenvolvimento com reuso. O processo Gerência de Ativos (Reusáveis) cobre os principais pontos sobre o processo de gerenciamento do repositório.
- c) traduzir os três processos do grupo de reuso da ISO 15504-5 para o formato de área de processo do CMMI-SE/SW para facilitar a utilização pela comunidade já acostumada com o CMMI-SE/SW. Esta tradução foi realizada primeiramente por Ferreira [2005] e depois revista e evoluída.
- d) identificar as áreas de processo do CMMI-SE/SW mais afetadas pela Engenharia de Software Baseada em Componentes e prover alterações com relações ao uso das mesmas.
- e) considerar o UML Components como a referência para um modelo de processo de desenvolvimento generalizado baseado em componentes, porque ele é simples, representativo e já é adotado por parceiros dos projetos.
- f) considerar as práticas dos três parceiros do projeto como referência inicial de práticas para Engenharia de Software Baseada em Componentes. Todos os três parceiros são reconhecidos como excelentes empresas, que já utilizam abordagens de CBD e melhoria de processo com CMMI-SE/SW e outros modelos.

A primeira idéia foi utilizar o OOSPICE como o Modelo de Capacidade do Processo. A decisão de não usar o OOSPICE foi tomada devido a três razões principais:

- a) O OOSPICE não é um modelo público, e não foi obtido o acesso nem mesmo direito para usá-lo, o acesso foi concedido somente a artigos e apresentações disponíveis no site do OOSPICE. Foi solicitado o acesso ao modelo completo, mas ele não foi concedido e nem a permissão do seu uso.
- b) Não foi intenção do projeto desenvolver um modelo de capacidade de processo, mas sim um conjunto de modelos de capacidade de processo a serem agregados com os atuais. Nesta visão com a aplicação do reuso e de componentes faz com que este modelo seja um facilitador à sua utilização pela indústria de software;
- c) É necessário ter o controle de toda a tecnologia do projeto com o objetivo de facilitar sua utilização por outros parceiros do projeto.



Os projetos COMP-GOV e FLO-PREF são projetos relativamente curtos, com recursos limitados, e seu foco principal é oferecer boas soluções prontas para a indústria de software em aplicações no domínio de governo eletrônico.

Do ponto de vista operacional, um projeto com dois anos foi planejado, para 2005 e 2006, com cinco fases:

Fase 1:

Estudo e revisão bibliográfica da Engenharia de Software Baseada em Componentes e de Modelos de Capacidade de Processo;

Fase 2:

Definição preliminar do Modelo de Capacidade de Processo Baseado em Componentes;

Fase 3:

Avaliações piloto em três empresas;

Fase 4:

Revisão do modelo baseada na experiência piloto e consolidação do Modelo de Capacidade de Processo Baseado em Componentes; e

Fase 5:

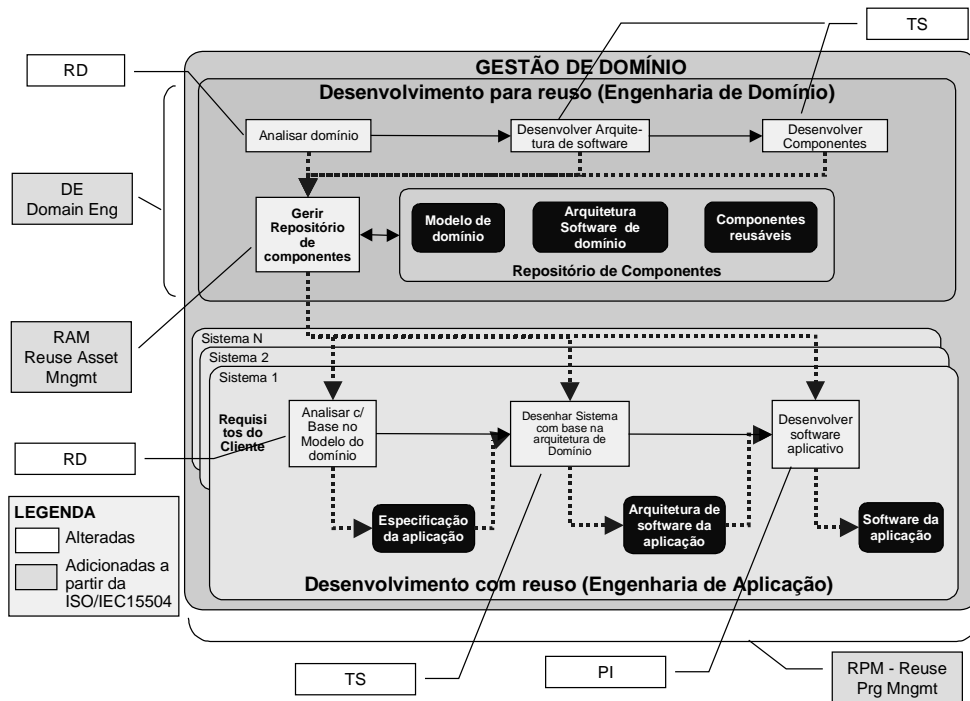
Avaliação aplicada em todos os desenvolvedores de componentes, repositórios e usuários do repositório.

5.1. Estudo e revisão bibliográfica

As duas primeiras fases do projeto foram realizadas em 2005. Um estudo amplo [Tsukumo 2005a] da literatura ([Cheesman e Daniels 2001], [Crnkovic e Larsson 2002], [Szipperski 1998] entre muitos outros) sobre CBD foi feito para determinar os aspectos que diferenciam seus processos dos “normais” e que devem ser enfatizados numa avaliação de processo. Por ser adotado pelos parceiros, foi dada especial atenção ao UML Components [Cheesman 2001]. Foram estudados também os modelos de capacidade de processo mais difundidos: CMMI-SE/SW [Chrissis et al. 2004], iCMM [Ibrahim et al. 2001], ISO/IEC 15504-5 [ISO/IEC 2006] e o MR-MPS do MPS.BR [SOFTEX 2005].

Após estes estudos, foi definido em documento do projeto [Tsukumo 2005b], o que fazer:

- a) Definir as áreas de processo mais afetadas pelo CBD e que deveriam sofrer alterações e incluir novas recomendações: Desenvolvimento de Requisitos, Solução Técnica e Integração de Produto; e
- b) incluir três novas áreas de processo correspondentes ao Grupo de Reuso definido pela ISO/IEC 15504-5: Gerência do Programa de Reuso, Gerência dos Ativos de Reuso e Engenharia de Domínio. O objetivo dessas mudanças foi evidenciar os principais aspectos do processo CBD comparado ao “normal”.



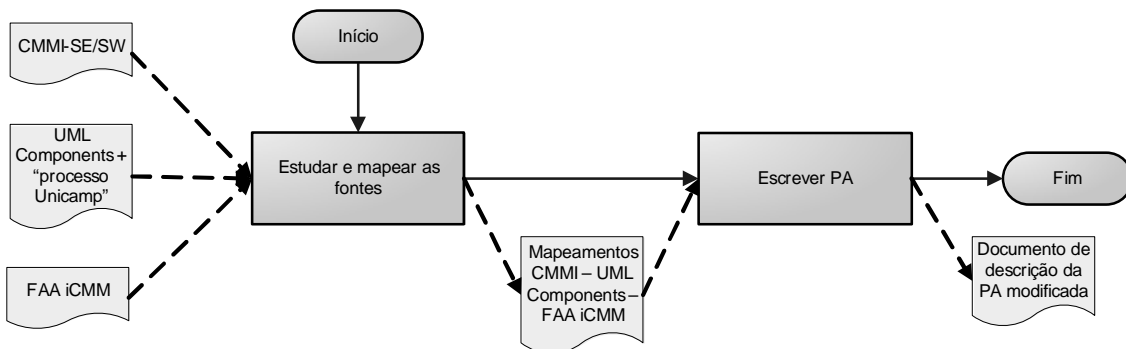
A Figura 2 mostra as áreas de processo do CMMI-SE/SW que foram consideradas para sofrerem alterações e os processos do grupo de reuso da ISO/IEC 15504-5 mapeadas no processo CBD.

Figura 2 - Mapeamento das áreas de processo alteradas e incluídas no processo CBD
Modificado a partir de [Foreman 96]

5.2. Modificações em áreas de processo do CMMI-SE/SW²

5.2.1. Metodologia

A modificação das áreas de processo (PAs) do CMMI-SE/SW se deu através de estudo do modelo FAA iCMM, do processo proposto pelo livro *UML Components* [Cheesman 2001] e do processo proposto pela Universidade Estadual de Campinas (UNICAMP). A Figura 3 apresenta as etapas para a elaboração das PA's.



² Devido à familiaridade para as pessoas que trabalham com o modelo CMMI SE/SW, são utilizadas, daqui para frente, as siglas em inglês para os elementos do modelo: PA (Process Area - Área de Processo); SG (Specific Goal - Objetivo Específico); SP (Specific Practice – Prática específica)



Figura 3 – Visão Geral da Metodologia

Primeiramente foi realizado um estudo de uma determinada área de processo do CMMI-SE/SW, com o objetivo de entender melhor os aspectos considerados pela mesma dentro de um contexto voltado ao desenvolvimento de software baseado em componentes e suas principais deficiências.

Em seguida foi realizado um estudo do processo de desenvolvimento baseado em componentes proposto pelo livro *UML Components* e pela Universidade Estadual de Campinas – UNICAMP. Foram consideradas neste estudo todas as atividades descritas pelo *UML Components* e pelo processo da UNICAMP relacionadas ao contexto abordado pela área de processo (PA) do CMMI-SE/SW analisada. Foram estudadas também as áreas de processo do iCMM correspondentes. Esses estudos foram registrados em mapeamentos entre os modelos, que permitiram definir as alterações necessárias, expressas em uma proposta representada graficamente, para facilitar a visualização e a compreensão das alterações propostas.

Um exemplo de representação gráfica de proposta de modificações a uma PA pode ser vista na figura 4.

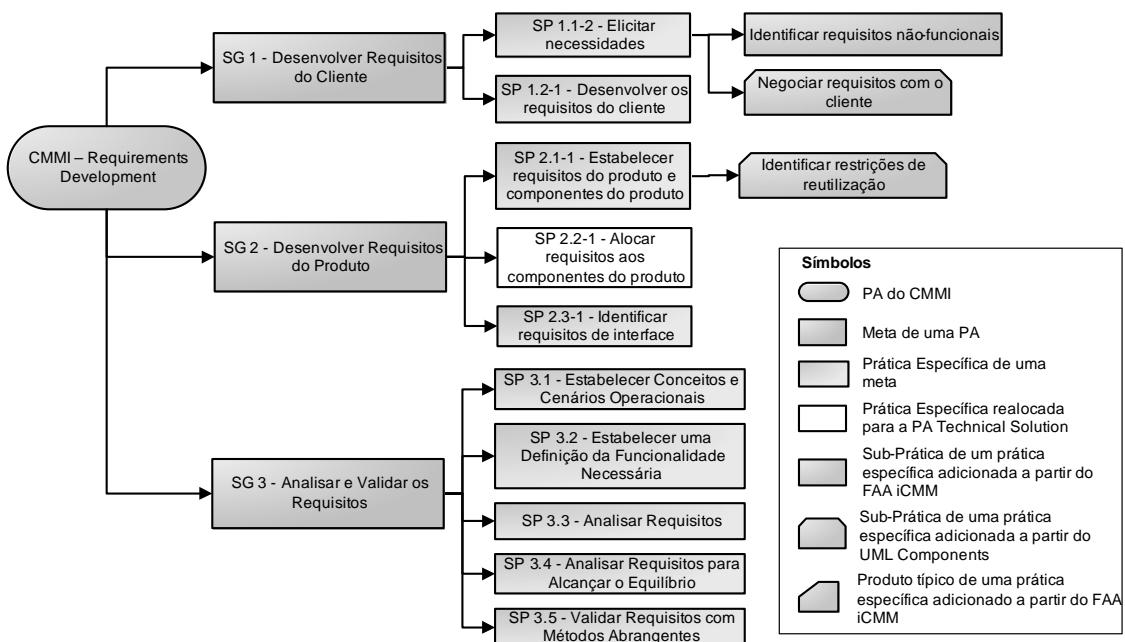


Figura 4 - Alteração proposta graficamente para a Área de Processo Desenvolvimento de Requisitos

Por fim, estas alterações propostas graficamente foram discutidas, revisadas, alteradas, quando necessário, e transferidas em forma de texto para o modelo proposto seguindo o padrão adotado pelas áreas de processo do CMMI-SE/SW.

5.2.2. Alterações

As alterações propostas têm por objetivo principal explicitar mais claramente as atividades relacionadas ao desenvolvimento baseado em componentes dentro do modelo CMMI-SE/SW. Isto implica em trazer o que estava nas entrelinhas do modelo CMMI-SE/SW para algo mais facilmente observável aos utilizadores do mesmo, além de adicionar aspectos não considerados anteriormente pelo modelo CMMI-SE/SW. As



principais alterações propostas para as três áreas de processo do CMMI-SE/SW estão descritas a seguir.

5.2.2.1. Desenvolvimento de Requisitos

A principal alteração da área de processo Desenvolvimento de Requisitos foi a adição no SG1, SP 1.1, da sub-prática ‘Negociar requisitos com o cliente’, para refletir a ação de reaproveitamento de componentes anteriormente adquiridos ou produzidos. Através desta os requisitos passam a ser re-avaliados com o cliente levando-se em consideração vantagens de se utilizar componentes já existentes no repositório. Na mesma prática específica foi adicionada a sub-prática ‘Identificar Requisitos não funcionais’ para destacar a importância deste tipo de requisitos.

No SG2, SP 2.1, foi adicionada a sub-prática ‘Identificar restrições de reutilização’. A prática específica ‘Alocar requisitos aos componentes do produto’ foi transferida para a área de processo ‘Solução Técnica’. A Tabela 1 mostra a estrutura proposta para a área de processo Desenvolvimento de Requisitos (RD), refletindo as mudanças citadas.

Tabela 1 – Estrutura proposta para a Área de Processo Desenvolvimento de Requisitos

Desenvolvimento de Requisitos	
SG 1 Desenvolver Requisitos do Cliente	
	SP 1.1 Elicitar Necessidades
	<i>Sub-prática Identificar Requisitos Não Funcionais (adicionada)</i>
	<i>Sub-prática Negociar Requisitos com o Cliente (adicionada)</i>
	SP 1.2 Desenvolver os Requisitos do Cliente
SG 2 Desenvolver Requisitos do Produto	
	SP 2.1 Estabelecer os Requisitos de Produtos e Componentes de Produtos
	<i>Sub-prática Identificar Restrições de Reutilização (adicionada)</i>
	SP 2.2 Alocar Requisitos aos Componentes do Produto (Realocada para PATS)
	SP 2.2 Identificar Requisitos de Interfaces
SG 3 Analisar e Validar os Requisitos	
	SP 3.1 Estabelecer Conceitos e Cenários Operacionais
	SP 3.2 Estabelecer uma Definição da Funcionalidade Necessária
	SP 3.3 Analisar Requisitos
	SP 3.4 Analisar Requisitos para Alcançar o Equilíbrio
	SP 3.5 Validar Requisitos com Métodos Abrangentes

Obs.: Nesta e nas outras tabelas, os textos em itálico representam as principais alterações realizadas.

5.2.2.2. Solução Técnica

A principal proposta de modificação para esta área de processo foi a divisão do Objetivo específico ‘Desenvolver *Design*’ nas metas ‘Definir Arquitetura’, ‘Desenvolver Especificações de Interface’ e ‘Desenvolver Especificações dos Componentes’, com a conseqüente modificação na ordenação e numeração das SG. O objetivo principal dessas alterações foi ressaltar no modelo proposto aspectos



fundamentais ao desenvolvimento de software baseado em componentes. Considerando mais detalhadamente os aspectos críticos relacionados ao projeto da arquitetura, especificação de interface e especificações dos próprios componentes.

Foram adicionadas várias sub-práticas aos objetivos específicos resultantes do desmembramento. Foi adicionada a prática específica 'Estabelecer Especificações da Interface' com as sub-práticas 'Avaliar restrições arquiteturais', 'Identificar arquiteturas iniciais'. A tabela 2 mostra a estrutura resultante para essa área de processo:

Tabela 2 – Resultado da Alteração sobre a Área de Processo Soluções Técnicas

Solução Técnica	
<i>SG 1 Definir Arquitetura (desmembrada da SG2 Desenvolver Design)</i>	
	<i>SP 1.1 Estabelecer Especificações da Arquitetura (adicionada)</i>
	<i>Sub-Prática Avaliar Restrições Arquiteturais (adicionada)</i>
	<i>Sub-Prática Identificar Arquiteturas Iniciais (adicionada)</i>
<i>SG 2 Selecionar Soluções de Componentes de Produtos (antiga SG1)</i>	
	<i>SP 2.1 Desenvolver Soluções Alternativas Detalhadas e Critérios de Seleção</i>
	<i>Sub-Prática Identificar Componentes Iniciais (adicionada)</i>
	<i>SP 2.2 Evoluir Conceitos Operacionais e Cenários</i>
	<i>SP 2.3 Selecionar Soluções de Componentes de Produtos</i>
	<i>SP 2.4 Executar Análises de Construção, Compra ou Reuso (Realocada)</i>
	<i>Produto Típico: Conjunto de COTS padronizados (adicionada)</i>
	<i>Produto Típico: Estratégia de não desenvolvimento de itens (adicionada)</i>
	<i>Produto Típico: Plano para evolução das versões dos produtos de prateleira</i>
<i>SG 3 Desenvolver Especificações de Interface (desmembrada da Meta Desenvolver Design)</i>	
	<i>SP 3.1 Criar o Design das Interfaces Utilizando os Critérios</i>
	<i>SP 3.2 Estabelecer Descrições de Interface</i>
	<i>Sub-Prática Requisitos de Interface (adicionada)</i>
	<i>Sub-Prática Especificação de Interface do Usuário (adicionada)</i>
	<i>Sub-Prática Especificação de Interface de Sub-Sistemas (adicionada)</i>
	<i>Sub-Prática Especificação de Interface de Componentes (adicionada)</i>
<i>SG 4 Desenvolver Especificações dos Componentes (desmembrada da Meta Desenvolver Design)</i>	
	<i>SP 4.1 Projetar o Produto ou Produzir o Componente</i>
	<i>Sub-Prática Completar os Serviços Oferecidos Parcialmente</i>
	<i>Sub-Prática Adaptar as Incompatibilidades</i>
	<i>SP 4.2 Alocar Requisitos aos Componentes do Produto (Realocada de RD-SP2.2)</i>
	<i>SP 4.3 Estabelecer um Pacote de Dados Técnicos</i>
<i>SG 5 Implementar o Design do Produto</i>	
	<i>SP 5.1 Implementar o Design</i>
	<i>SP 5.2 Desenvolver a Documentação de Suporte do Produto</i>
	<i>Sub-Prática Gerar um Repositório de Dados do Projeto</i>



5.2.2.3. Integração de Produtos

A principal alteração da área de processo Integração de Produtos foi a importância dada aos conectores. Sub-práticas e produtos típicos foram adicionados com o objetivo de dar uma maior importância aos conectores, pois estes são os únicos componentes do sistema que possuem o conhecimento de seus fluxos interativos. Dessa forma, eles são considerados os locais ideais para a implementação dos requisitos de qualidade do sistema, tais como a tolerância à falhas, distribuição e disponibilidade.

Sub-práticas: Foram adicionadas as sub-práticas ‘Assegurar confiabilidade dos conectores’ e ‘Medir os atributos de qualidade do componente através dos conectores’; Produtos Típicos: Foram adicionados os produtos típicos ‘Análise do reuso de interface de componentes’ e ‘Relatório de teste de instalação de integração’. A Tabela 3 mostra a estrutura resultante dessa área de processo.

Tabela 3 – Resultado da Alteração sobre a Área de Integração de Produtos

Integração de Produtos	
SG 1 Preparar para a Integração do Produto	
	SP 1.1 Determinar a Sequência de Integração
	SP 1.2 Estabelecer o Ambiente de Integração do Produto
	SP 1.3 Estabelecer os Procedimentos e Critérios de Integração do Produto
SG 2 Assegurar a Compatibilidade das Interfaces	
	SP 2.1 Revisar as Descrições das Interfaces quanto a sua Completitude
	<i>Produto Típico Análise do reuso de interfaces de componentes (adicionada)</i>
	SP 2.2 Gerenciar as Interfaces
SG 3 Montar os Componentes do Produto e Entregar o Produto	
	SP 3.1 Confirmar que os Componentes do Produto estão Prontos para Integração
	SP 3.2 Montar os Componentes do Produto
	<i>Sub-Prática Assegurar Confiabilidade dos Conectores(adicionada)</i>
	SP 3.3 Avaliar os Componentes do Produto Montados
	<i>Sub-Prática Medir os Atributos de Qualidade Através dos Conectores (adicionada)</i>
	SP 3.4 Empacotar e Entregar o Produto ou Componente do Produto

5.3. Novas áreas de processo para CBD

5.3.1. Metodologia

A organização das PAs do CMMI-SE/SW segue um padrão diverso da organização dos “Processos” da ISO/IEC 15504-5. Uma primeira tentativa de transpor os processos do grupo de reuso da ISO/IEC 15504-5 para o formato do CMMI-SE/SW [Ferreira 2005] colocava apenas um “Objetivo Específico” para cada PA e transformava cada *Base Practice* em uma “Prática Específica”. Essa organização não nos pareceu adequada e preferimos agrupar em “Objetivos específicos”, as práticas relativas à definição, à operação e, no caso da PA de Gerência do Programa de Reuso, à monitoração.

5.3.2. Descrição das áreas de processo

As áreas de processos inseridas a partir da ISO/IEC 15504-5 foram definidas



utilizando-se a sintaxe do CMMI-SE/SW como descritas abaixo.

Nas descrições foram omitidas as sub-práticas e os produtos típicos de trabalho existentes dentro de cada uma das práticas específicas adicionadas.

5.3.2.1. Gerência do Programa de Reuso (RPM)

O propósito da área de processo Gerência do Programa de Reuso (RPM) é planejar, estabelecer, gerenciar, controlar e monitorar o programa de reuso da organização e sistematicamente explorar oportunidades de reuso. A Tabela 4 mostra a estrutura resultante dessa área de processo .

Tabela 4 – Área de Processo de Gerência de Programa de Reuso (RPM)

Gerência do Programa de Reuso	
SG 1 Definir estratégia de reuso da organização	
	SP 1.1 Identificar domínios para as potenciais oportunidades de reuso
	SP 1.2 Avaliar a capacidade de reuso da organização
	SP 1.3 Avaliar o potencial de reuso de cada domínio
SG 2 Implementar estratégia de reuso da organização	
	SP 2.1 Avaliar as propostas de reuso para certificar-se que o produto reusado é apropriado para a aplicação proposta
	SP 2.2 Estabelecer formas de comunicação
SG 3 Monitorar e avaliar o programa de reuso	
	SP 3.1 Monitorar o programa de reuso
	SP 3.2 Analisar questões
	SP 3.3 Tomar ações corretivas
	SP 3.4 Gerenciar as ações corretivas

5.3.2.1. Gerência de Ativos Reusáveis (RAM)

O propósito da área de processo Gerência de Ativos Reusáveis (RAM) é gerenciar os ativos que podem ser reusados em diferentes projetos, sendo o ativo gerenciado desde a sua concepção, passando pelas alterações até a retirada definitiva deste de produção e do repositório. A Gerência de Ativos Reusáveis é responsável por adquirir, avaliar e organizar artefatos produzidos durante a engenharia de domínio. A Tabela 5 mostra a estrutura resultante dessa área de processo .

Tabela 5 – Área de Processo de Gerência de Ativos reusáveis (RAM)

Gerência de Ativos Reusáveis	
SG 1 Definir estratégia de gerência de ativos reusáveis	
	SP 1.1 Estabelecer um esquema de classificação dos ativos de software reusável
	SP 1.2 Definir um critério de aceitação do ativo, certificação e retirada dos ativos de software do repositório
	SP 1.3 Definir um mecanismo de armazenamento e acesso ao ativo
	SP 1.4 Documentar a estratégia de gerência de ativos
SG 2 Operar repositório de ativos reusáveis	
	SP 2.1 Disponibilizar mecanismo de armazenamento, registro, recuperação de ativos e comunicação
	SP 2.2 Identificar e/ou gerar ativos de software reusáveis



SP 2.3 Aceitar ativos reusáveis
SP 2.4 Disponibilizar ativos e registrar a sua utilização
SP 2.5 Gerenciar e controlar as mudanças dos ativos
SP 2.6 Retirar definitivamente os ativos

5.3.2.1. Engenharia de Domínio (DE)

O propósito da área de processo Engenharia de Domínio (DE) é capturar, organizar e representar conhecimento sobre um domínio e produzir ativos reutilizáveis que podem ser aplicados para produzir uma família de sistemas no domínio desenvolvendo e mantendo modelos de domínios, arquiteturas de domínio e ativos para o domínio. A Tabela 6 mostra a estrutura resultante dessa área de processo .

Tabela 6 – Área de Processo de Engenharia de Domínio (DE)

Engenharia de Domínio	
SG 1 Definir características do domínio	
	SP 1.1 Selecionar representação do domínio
	SP 1.2 Desenvolver modelos do domínio
SG 2 Definir necessidades de domínio	
	SP 2.1 Definir necessidades de ativos de domínio
	SP 2.2 Definir necessidades de arquitetura de domínio

A Figura 5 mostra o relacionamento entre as áreas de processo acima descritas.

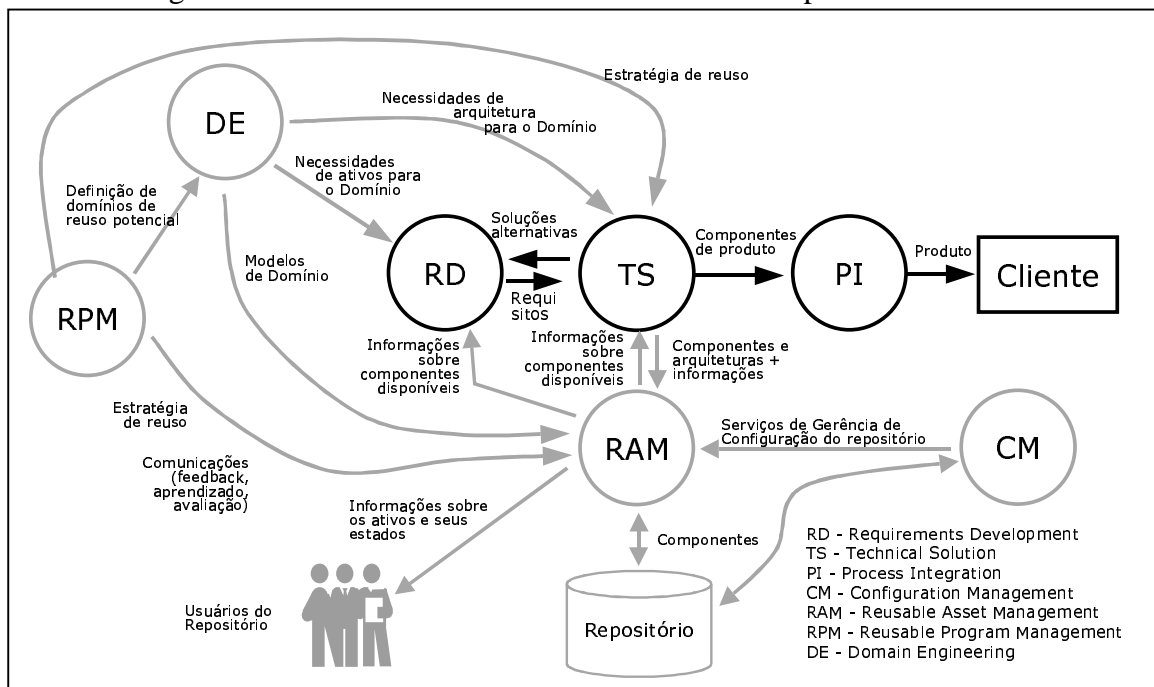


Figura 5: Relacionamento entre as áreas de processo para Desenvolvimento Baseado em Componentes



6. Conclusão

Este artigo descreveu uma estratégia para a especialização de modelos de capacidade de processo de software para o contexto específico da melhoria de processo de desenvolvimento de software baseado em componentes, e uma utilização exemplo desta estratégia para extensão do modelo CMMI-SE/SW v1.1 com a especialização de algumas de suas áreas de processo e a inclusão dos processos do grupo de reuso da ISO/IEC 15504-5.

A principal contribuição desta estratégia para o campo de estudos da qualidade de software em geral e da melhoria de processo de software em particular, é prover uma orientação para a especialização de modelos de capacidade de processo para uma área, no caso desenvolvimento de software baseado em componentes, com potencial de impacto na qualidade e produtividade de software.

A descrição da utilização exemplo foi mais extensa do que a descrição da estratégia porque além de mostrar como a estratégia foi utilizada em uma necessidade real também serve para esclarecer mais detalhes da estratégia e como uma prova de conceito da estratégia. Esta prova de conceito deve ser considerada como uma validação parcial porque a uma validação mais completa será realizada com a utilização deste exemplo. Esta utilização será realizada em 2006 nas fases 3, 4 e 5.

Agradecimentos

A utilização exemplo descrita nesse artigo é parte dos projetos COMP-GOV e FLO-PREF e, como tal, é financiada pela chamada MCT/FINEP/Ação Transversal - Biblioteca de Componentes - 05/2004.

Referências

- Cheesman, J.; Daniels, J. (2001) - UML Components: A Simple Process for Specifying Component-Based Software – Addison-Wesley.
- Chrissis, M.B.; Konrad M.; Shrum, S.(2004) – “CMMI – Guidelines for Process Integration and Product Improvement” – SEI Series in Software Engineering – Addison-Wesley.
- Crnkovic, I.; Larsson, L. (editors) (2002) – Building Reliable Component-Based Software Systems – Artech House.
- Ferreira, Jean de S. (2005) Descrevendo os processos do grupo de reuso da ISO/IEC 15504-5 para o padrão do CMMI, Monografia de conclusão do curso de especialização de Qualidade no Desenvolvimento de Software da Faculdade Senac de Ciências Exatas e Tecnologia.
- FINEP (2004) – Financiadora de estudos e projeto – Ministério da Ciência e Tecnologia – Governo do Brasil - Chamada pública mct/finep/ação transversal -biblioteca de componentes - 05/2004 - Seleção pública de propostas para apoio a projetos de inovação visando a constituição de uma biblioteca compartilhada de componentes para o domínio de aplicação “governo eletrônico”
- Foreman, J. (1996) Product Line Based Software Development- Significant Results, Future Challenges. Software Technology Conference, Salt Lake City, UT, April 23.



- Ibrahim, Linda, B. Bradford, D. Cole, L. LaBruyere, H. Leinneweber, D. Piszczek, N. Reed, M. Rymond, D. Smith, M. Virga and C. Wells (2001) , The Federal Aviation Administration Integrated Capability Maturity Model® (FAA-iCMM®), Version 2.0, An Integrated Capability Maturity Model for Enterprise-wide Improvement, Technical Report at The Federal Aviation Administration, 480 pages.
- ISO/IEC 15504-2 (2003) ISO/IEC 15504-2 - Information Technology - Software Engineering - Process Assessment - Part 2: Performing An Assessment - The International Organization for Standardization and the International Electrotechnical Commission.
- ISO/IEC 15504-5 (2006) “ISO/IEC 15504-5 - Information Technology - Process Assessment - Part 5: An exemplar Process Assessment Model” - The International Organization for Standardization and the International Electrotechnical Commission.
- MCT (2005) - Ministério da Ciência e Tecnologia; Sociedade para Promoção da Excelência do Software Brasileiro – SOFTEX e Departamento de Política Científica e Tecnológica - DPCT/UNICAMP – “Estratégia Nacional para Componentes de Software”.
- Salviano, Clênio F.(2006) , Uma proposta orientada a perfis de capacidade de processo para evolução da melhoria de processo de software, Tese de doutorado, Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas (FEEC-Unicamp).
- SOFTEX (2005) MPS.BR – Melhoria de Processo de Software Brasileiro – Guia Geral (versão 1.0).
- Stallinger, F., A. Dorling, T. Rout, B. Henderson-Sellers, B. Lefever (2002) Software Process Improvement for Component-Based Software Engineering: An Introduction to the OOSPICE Project, Proceedings of the 28th EUROMICRO Conference, September 4-6, 2002, Dortmund, Germany, IEEE Computer Society, Los Alamos, CA.
- Sziperski, C. (1998) – Component Software – Beyond Object-Oriented Programming – Addison-Wesley.
- Tsukumo A. N. et al. (2005a) - Revisão Bibliográfica sobre Processo baseado em componentes (Documento de projeto – COMP-GOV e FLO-PREF).
- Tsukumo A. N. et al. (2005b) - Especificação de Modelo de Referência de Processo para Desenvolvimento Baseado em Componentes – PRM.CBD (Documento de projeto – COMP-GOV e FLO-PREF).